حسین بیگی – نسخه سند ۲۰۷-۱

فیلتر های Opencv

Changing Colorspaces

بیشتر از ۱۵۰ حالت برای تبدیل رنگ‌ها وجود دارد که معروف ترین آن‌ها BGR → GARY & BGR → HSV میباشد

برای استفاده از این روش از تابع**cv۲.cvtColor(), cv۲.inRange()** استفاده میشودبرای مثال

```
import cv2
cv2.cvtColor(input_image,flag)
```

flag = نوع تبدیل میباشد که برای نمونه

cv2.COLOR_BGR2GRAY

cv2.COLOR_BGR2HSV

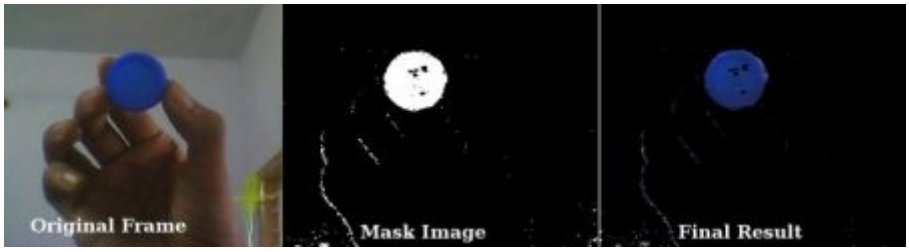
cv2.COLOR_BGR2RGB

میتوان استفاده کرد

پیدا کردن یک توپ آبی با استفاده از تبدیل رنگ‌ها HSV

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

while(1):

    # Take each frame
    _, frame = cap.read()

    # Convert BGR to HSV
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # define range of blue color in HSV
    lower_blue = np.array([110,50,50])
```

```
    upper_blue = np.array([130,255,255])

    # Threshold the HSV image to get only blue colors
    mask = cv2.inRange(hsv, lower_blue, upper_blue)

    # Bitwise-AND mask and original image
    res = cv2.bitwise_and(frame,frame, mask= mask)

    cv2.imshow('frame',frame)
    cv2.imshow('mask',mask)
    cv2.imshow('res',res)
    k = cv2.waitKey(5) & 0xFF
    if k == 27:
        break

cv2.destroyAllWindows()
```



حسین بیگی – نسخه سند ۲۰۷-۱

Image Blurring (Image Smoothing) اطلاعات بیشتر

1 – متد Averaging

این الگو میانگین تمام پیکسل های مشخص شده به اندازه هسته(Kernel) را گرفته و پیکسل وسط را با آن جایگزین میکند

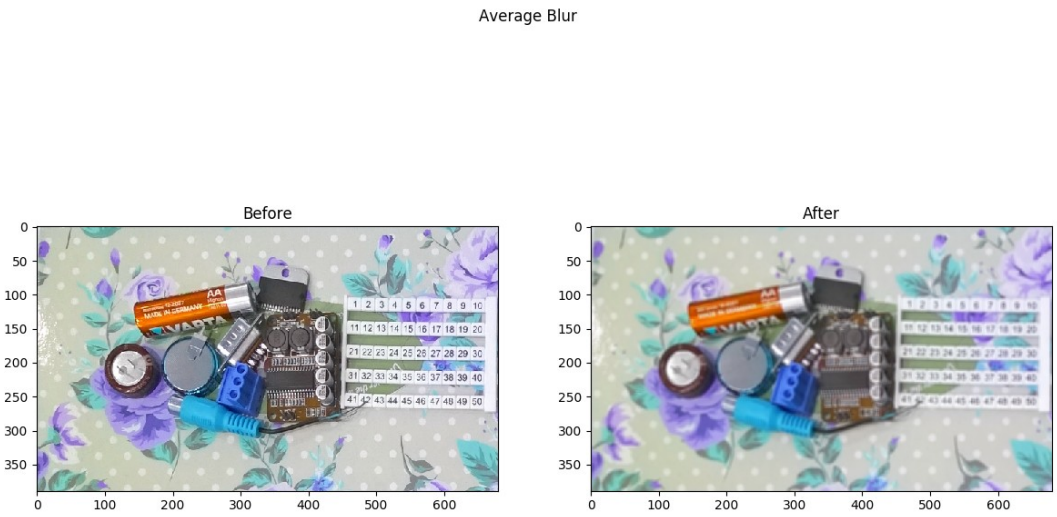
$$K = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

نمونه یک هسته سه در سه

کد:

```
Import cv2
import matplotlib.pyplot as plt
import numpy as np
src = cv2.imread("DATA/filter_sample.jpg")
##  kernel = (5,5)


filtername = "Average Blur"
gb = cv2.blur(src,(5,5),0)
plt.subplot(121)
plt.imshow(src)
plt.title("Before")
plt.subplot(122)
```

```
plt.imshow(gb)

plt.title("After")

plt.suptitle(filtername)

plt.show()
```

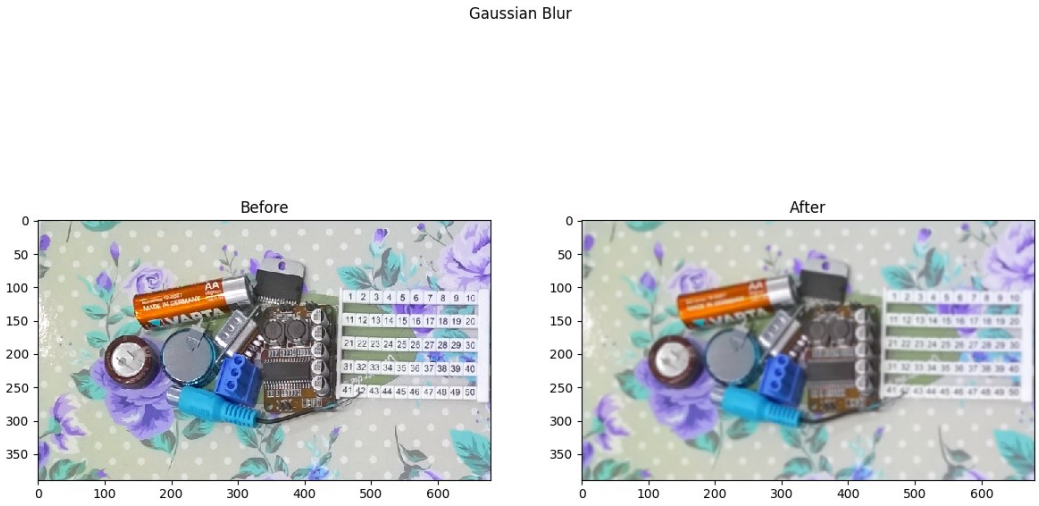


Gaussian Blurring -2

این فیلتر بر اساس تابع Guassian کار میکند اطلاعات بیشتر

```
#####Gaussian Filter#####

filtername = "Gaussian Blur"

gb = cv2.GaussianBlur(src,(9,9),0)

plt.subplot(121)

plt.imshow(src)
```

```
plt.title("Before")

plt.subplot(122)

plt.imshow(gb)

plt.title("After")

plt.suptitle(filtername)

plt.show()
```

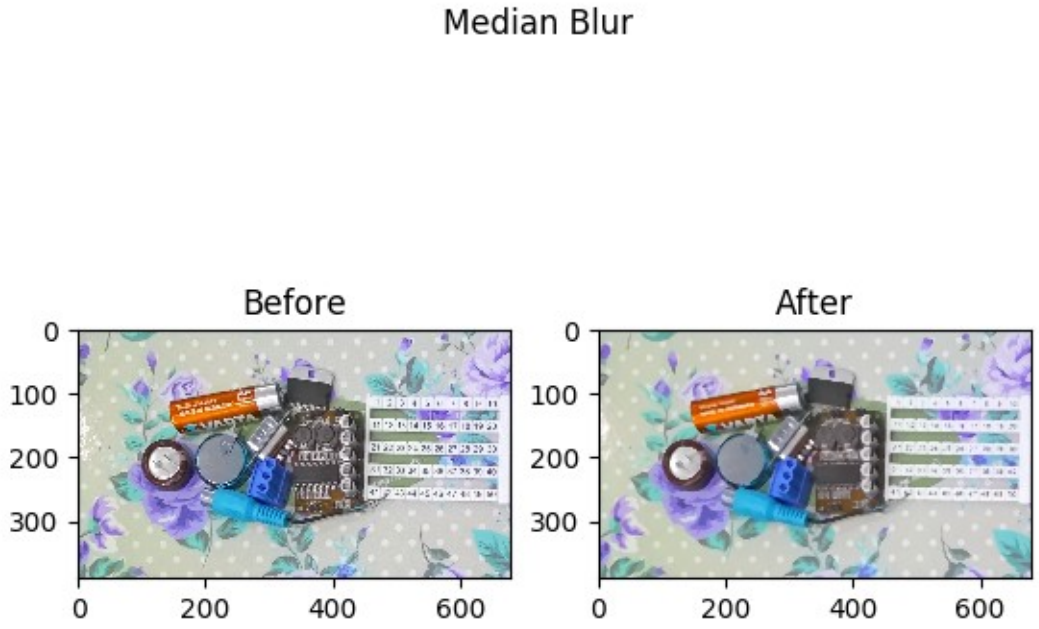


۳-متد Median Blurring

این متد میانه پیکسل های به اندازه مشخص شده در هسته را گرفته و در پیکسل وسط قرار میدهد

این روش برای از بین برد نویز هایی مانند نمکی و فلفلی (Salt and Pepper) بسیارخوب میباشد

```
#####Median Filter#####
filtername = "Median Blur"
gb = cv2.medianBlur(src,5)
plt.subplot(121)
plt.imshow(src)
plt.title("Before")
plt.subplot(122)
plt.imshow(gb)
plt.title("After")
plt.suptitle(filtername)
plt.show()
```


Median Blur
Before
After
0
100
200
300
0
200
400
600

حسین بیگی – نسخه سند ۲۰۷-۱

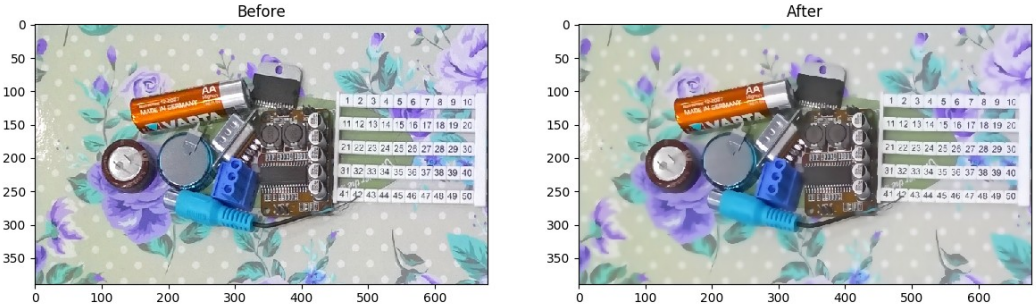
۴- متد Bilateral Filtering

میتوان گفت که این روش بهترین روش برای حذف نویز بوده که همچنین باعث حفظ لبه ها هم میشود.

```
#####Bilateral Filter#####
filtername = "Bilateral Blur"
gb = cv2.bilateralFilter(src,9,75,75)
plt.subplot(121)
plt.imshow(src)
plt.title("Before")
plt.subplot(122)
plt.imshow(gb)
plt.title("After")
plt.suptitle(filtername)
plt.show()
```

حسین بیگی – نسخه سند ۲۰۷-۱

Bilateral Blur



حسین بیگی – نسخه سند ۲۰۷-۱

تبدیل های ریخت شناسی **Morphological Transformations** اطلاعات بیشتر

این تبدیل ها به ساختار شکل اصلی کار دارد و تأثیر میگذارد و عموما در عکس هایی استفاده می‌شود که باینری باشد در حقیقت دو تبدیل اصلی Erosion و Dilation میباشد

* تمام عکس‌ها و کد های این بخش از سایت اصلی و سند آموزشی خود Opencv گرفته شده است

۱-Erosion

این فیلتر سعی میکند که همیشه جلوزمینه (Foreground) را سفید نگه دارد

کد:

```
import cv2
import numpy as np
img = cv2.imread('j.png',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
```

با اعمال فیلتر

عکس اصلی

2-Dilation

این روش دقیقاً برعکس روش Erosion

```
dilation = cv2.dilate(img,kernel,iterations = 1)
```

۳-Opening

این الگوریتم ترکیبی از دو الگورتیم بالا بوده و در مرحله اول Dilation و در مرحله دوم Erosion میباشد که برای حذف نویز های بیرونی بسیار خوب میباشد

```
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```

۴- Closing

این روش دقیقاً بر عکس روش بالا بوده و برای حذف نویز های داخلی بسیار عالی میباشد

```
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```

۵- Morphological Gradient

این روش از بین دو روش Dilation و Erosion

```
gradient = cv2.morphologyEx(img, cv2.MORPH_GRADIENT, kernel)
```

۶- Top Hat

این روش بین عکس اصلی و Opening یک عکس هست!

```
tophat = cv2.morphologyEx(img, cv2.MORPH_TOPHAT, kernel)
```

۷-Black Hat

این روش بین عکس اصلی و Closing میباشد

```
blackhat = cv2.morphologyEx(img, cv2.MORPH_BLACKHAT, kernel)
```